OIPE IAP43
APR 19 2006
PATENT & TRADEMARK OFFICE

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Re: Appeal to the Board of Patent Appeals and Interferences

In re PATENT application of ACHARYA — Group Art Unit: 2155

Application No. 09/905,080 — Examiner: NGUYEN, Thu Ha T

Filed: July 16, 2001

Title: Arrangement for Reducing Application Execution Based on a Determined Lack of Flow Control Credits for a Network Channel — Docket : 95-508

Date: April 19, 2006

Commissioner of Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

1 [ ] **NOTICE OF APPEAL**: Applicant hereby appeals to the Board of Patent Appeals and Interferences from the decision (not Advisory Action) dated July 28, 2005 of the Examiner twice/finally rejecting claims

2 [X] **BRIEF** on appeal in this application attached.

3 [ ] An **ORAL HEARING** is respectfully requested under Rule 194 (due two months after Examiner's Answer unextendable).

4 [ ] Reply Brief is attached (due two months after Examiner's Answer -- unextendable).

5. **FEE CALCULATION:**		Large/Small Entity	
If box 1 above is X'd, see box 12 below first and decide: .... enter		$500/250*	$
If box 2 above is X'd, see box 12 below first and decide: .... enter		$500/250*	$ 500.00
If box 3 above is X'd, see box 12 below first and decide: .... enter		$1000/500*	$
If box 4 above is X'd, ... enter nothing		- 0 - (no fee)	
6. Original due date: March 19, 2006			
7. **Petition is hereby made** to extend the original due date to cover the date this response is filed for which the requisite fee is attached (1 mo) (2mos) (3mos) (4mos)	$120 $450 $1020 $1590	+ $120	
8. Enter any previous extension fee paid [ ] previously since above original due date (item 6); [ ] with concurrently filed amendment		-	
9. **Subtract line8 from line7 and enter: Total Extension Fee**			+$120
10. **TOTAL FEE ATTACHED =**			**$ 620**

11. [ ] *Fee NOT required if/since paid in prior appeal in which the Board of Patent Appeals and Interferences did not render a decision on the merits.

CHARGE STATEMENT: The Commissioner is hereby authorized to charge any fee specifically authorized hereafter, or any missing or insufficient fee(s) filed, or asserted to be filed, or which should have been filed herewith or concerning any paper filed hereafter, and which may be required under Rules 16-18 (missing or insufficient fee only) now or hereafter relative to this application and the resulting Official document under Rule 20, or credit any overpayment, to our Account/Order No. 50-0687 / 95-508 for which purpose a duplicate copy of this sheet is attached. **This CHARGE STATEMENT does not authorize charge of the issue fee until/unless an issue fee transmittal form is filed**

Atty: [signature]
Leon R. Turkevich
Reg. No. 34,035
Tel: (202) 261.1000

Customer No. 20736

Docket No.: 95-508

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of :
:
ACHARYA :
:
Serial No.: 09/905,080 : Group Art Unit: 2155
:
Filed: July 16, 2001 : Examiner: NGUYEN, Thu Ha T

For: ARRANGEMENT FOR REDUCING APPLICATION EXECUTION BASED ON A DETERMINED LACK OF FLOW CONTROL CREDITS FOR A NETWORK CHANNEL

MAIL STOP: APPEAL BRIEF – PATENTS
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF

Sir:

This is an appeal from the final rejection of claims 1-13 in the above-identified patent application.

This Appeal Brief is submitted as required by 37 C.F.R. §41.37. Appellant concurrently submits a 1-month petition for extension of time in accordance with 37 C.F.R. §41.37(e) and 37 CFR §1.136.

1. Real Party in Interest:

This application is assigned to Advanced Micro Devices, Inc., the real party of interest.

2. Related Appeals and Interferences:

There are no other appeals or interferences known to Appellant that will directly affect or

be directly affected by or have a bearing on the Board's decision in the pending appeal.

3. Status of Claims:

Claims 1-13 are pending in this application. Claims 1-13 stand rejected by the Examiner, and claims 1-13 are appealed.

4. Status of any Amendment File Subsequent to Final Rejection:

No Amendment was filed in response to the Final Rejection. A Response to the Final Rejection was filed on September 21, 2005.

5. Summary of Claimed Subject Matter:

The claimed subject matter includes independent claims 1 and 7 and dependent claims 2-6 and 8-13. Independent claims 1 and 7 each specify a network node (illustrated in Fig. 2, page 4, lines 9-11 and 19-21) that selectively outputs a prescribed data stream based on a determined availability of flow control resources: claim 1 specifies a method "in a network node", and claim 7 specifies "[a] network node." The claimed network node illustrated in Fig. 2 includes a network interface such as a Host Channel Adapter (HCA) 120, a memory controller 130, system memory 48, and a processor 110 that executes application resources 140a and 140b. The Host Channel Adapter (HCA) (e.g., 12 of Fig. 1 or 120 of Fig. 2) provides an interface connection to the InfiniBand™ network (e.g., 10 of Fig. 1) (see, e.g., page 2, lines 5-9, page 4, lines 27-31).

Each of the independent claims 1 and 7 also specify that the network interface 120 in the network node (illustrated in Fig. 2) detects (step 200 of Fig. 3) a depletion of flow control resources representing a depletion of network bandwidth for a prescribed data stream (page 5, lines 1-3 and 15-16): the network interface (120 of Fig. 2) outputs (step 210 of Fig. 3) a data flow interruption request based on the detected depletion of flow control resources. The processor 110 in the network node reduces the prescribed data stream, based on the data flow interruption request, by reducing execution of a prescribed application resource (steps 220, 230, and 240 of Fig. 3; page 4, line 27 to page 5, line 11 and page 5, lines 19-31).

Hence, the supply of data for the data packets is reduced based on the reduced execution of the prescribed application resource, enabling flow control protocols to be maintained while preserving processor and memory resources for other application resources (see page 3, lines 8-10). Moreover, network congestion can be resolved without the necessity of wasting resources by dropping packets generated by the processor and stored in a system memory (page 3, lines 19-24). Hence, processor resources and system memory can be redirected to unaffected application resources, optimizing efficiency in the network node (page 11, lines 23-26).

Hence, independent claim 1 specifies a method in a network node (illustrated in Fig. 2, page 4, lines 9-11 and 19-21). The method includes detecting by a network interface in the network node (120 of Fig. 2, page 4, lines 19-21) a depletion of flow control resources representing a depletion of network bandwidth for a prescribed data stream (200 of Fig. 3, page 5, lines 1-3 and 15-16), and outputting (210 of Fig. 3) by the network interface a data flow interruption request based on the detected depletion of flow control resources (page 5, lines 1-3 and 15-19). The method also includes reducing, by a processor in the network node (110 of Fig. 2, page 4, lines 19-26) and based on the data flow interruption request, the prescribed data stream by reducing execution of a prescribed application resource (e.g., 140a of Fig. 2) configured for generating the prescribed data stream (220, 230, and 240 of Fig. 3, page 4, lines 27 to page 5, line 11; page 5, lines 19-31).

Claim 2 adds to the method of claim 1, wherein the network interface is configured for outputting the prescribed data stream according to InfiniBand™ protocol (page 4, lines 24-26), the detecting step including detecting depletion of flow control credits, as the flow control resources, for a prescribed virtual lane (page 5, lines 15-16).

Claim 3 adds to method of claim 2, wherein the outputting step includes outputting the data flow interruption request to a memory controller (130 of Fig. 2, page 4, lines 19-21) configured for controlling access to system memory resources, the memory controller rendering unavailable the system memory resources for the prescribed application resource in response to reception of the data flow interruption request (220 of Fig. 3, page 5, lines 3-7 and 19-21).

Claim 4 adds to the method of claim 3, wherein the reducing step (by processor 110 of

Fig. 2) includes halting execution of the prescribed application resource (140a of Fig. 2), based on a determined unavailability of the system memory resources (230 and 240 of Fig. 3, page 5, lines 22-31).

Claim 5 adds to the method of claim 4 by outputting by the network interface a resume data flow request based on a detected replenishment of the flow control resources for the prescribed data stream (250 and 260 of Fig. 3, page 5, line 32 to page 6, line 2).

Claim 6 adds to the method of claim 5 by resuming execution of the prescribed application resource based on the resume data flow request (262 of Fig. 3; page 5, line 33 to page 6, line 3).

Independent claim 7 specifies a network node (illustrated in Fig. 2, page 4, lines 9-11 and 19-21). The network node comprises a network interface (120 of Fig. 2) configured for detecting a depletion of flow control resources representing a depletion of network bandwidth for a prescribed data stream (200 of Fig. 3, page 5, lines 1-3 and 15-16), the network interface configured for outputting a data flow interruption request based on the detected depletion of flow control resources (210 of Fig. 3, page 5, lines 1-3 and 15-19). The network node also comprises a processor (110 of Fig. 2) configured for executing a prescribed application resource (e.g., 140a of Fig. 2) for generation of the prescribed data stream (4:22-27), the processor configured for reducing the prescribed data stream by reducing execution of the prescribed application resource, based on the data flow interruption request (220, 230 and 240 of Fig. 3, page 4, line 27 to page 5, line 11; page 5, lines 19-31).

Claim 8 adds to the network node of claim 7 a memory controller (130 of Fig. 2) configured for controlling access to system memory resources (page 4, lines 19-21), the memory controller configured for rendering unavailable the system memory resources for the prescribed application resource in response to reception of the data flow interruption request (22 of Fig. 3, page 5, lines 3-7 and 19-21).

Claim 9 adds to the network node of claim 8, wherein the processor is configured for reducing the execution (240 of Fig. 3) of the prescribed application resource (140a of Fig. 2) based on detecting (230 of Fig. 3) the unavailability of the system memory resources (page 5,

lines 22-31).

Claim 10 adds to the network node of claim 9, wherein the network interface is configured for outputting a resume data flow request (260 of Fig. 3) based on a detected replenishment of the flow control resources for the prescribed data stream (250 of Fig. 3, page 5, line 32 to page 6, line 2).

Claim 11 adds to the network node of claim 10, wherein the processor is configured for resuming execution of the prescribed application resource based on the resume data flow request (262 of Fig. 3, page 5, line 33 to page 6, line 3).

Claim 12 adds to the network node of claim 7, wherein the network interface is configured for outputting the prescribed data stream according to InfiniBand™ protocol (page 4, lines 24-26), the network interface configured for detecting depletion of flow control credits, as the flow control resources, for a prescribed virtual lane (page 5, lines 15-16).

Claim 13 adds to the method of claim 1, wherein the outputting step includes outputting the data flow interruption request to a memory controller (130 of Fig. 2, page 4, lines 19-21) in the network node and that is configured for controlling access to system memory resources in the network node, the memory controller rendering unavailable the system memory resources for the prescribed application resource in response to reception of the data flow interruption request (220 of Fig. 3, page 3, lines 3-7 and 19-21).

6. <u>Grounds of Rejection to be Reviewed on Appeal</u>:

A. Whether independent claims 1 and 7 are unpatentable under 35 U.S.C. §102(e) as having been anticipated in view of U.S. Patent Publication No. 2002/0085493 by Pekkala et al.

B. Whether dependent claims 3, 8, and 13 are unpatentable under 35 U.S.C. §102(e) as having been anticipated in view of U.S. Patent Publication No. 2002/0085493 by Pekkala et al.

7. Arguments:

A. **Claims 1 and 7 are not anticipated 35 U.S.C. §102(e) in view of Pekkala et al.**

In the Final Office Action, the Examiner rejected independent claims 1 and 7 under 35 USC §102(e) in view of Pekkala et al. Claims 1 and 7 are neither anticipated nor rendered obvious by Pekkala et al. for the following reasons.

A1. **Pekkala et al. Does Not Disclose or Suggest the Claimed Data Flow Interruption Request**

Each of the independent claims 1 and 7 specify that the network interface outputs "a data flow interruption request based on the detected depletion of flow control resources". Each of the independent claims also specify that the depletion of flow control resources "represents a depletion of network bandwidth for a prescribed data stream."

As described in detail below with respect to argument A2, the processor reduces the prescribed data stream by reducing execution of a prescribed application resource (configured for generating the prescribed data stream) based on the data flow interruption request. In other words, each of the independent claims 1 and 7 specify an implementation of flow control (by reducing execution of prescribed application resources based on the data flow interruption request), as opposed to flow control *per se*.

Hence, claims 1 and 7 explicitly distinguish between the notoriously well known procedure of "flow control" and the output of a "data flow interruption request". In other words, the claimed "data flow interruption request" cannot be so broadly construed as to encompass a "flow control packet" between distinct network nodes (i.e., link partners), because such an

interpretation would be unreasonable in view of the specification and the interpretation one skilled in the art would reach.[1]

The specification provides numerous references to the use of flow control, as described by the InfiniBand™ Architecture Specification (Release 1.0), for controlling the transmission of packets via an InfiniBand™ link:

> the InfiniBand™ Architecture Specification describes a flow control arrangement, where each virtual lane (VL) has a corresponding number of flow control credits. However, conventional approaches to implementing flow control result in interruption of data flows, dropped packets, etc., resulting in poor utilization of memory and processor resources.
>
> (Page 2, lines 23-26).
>
> There is a need for an arrangement that enables a computing node, configured for outputting data onto a data network according to prescribed flow control protocols, to execute applications configured for outputting data onto the data network in an efficient and economical manner.
>
> These and other needs are attained by the present invention, where a network node includes a network interface, a system memory, a memory controller configured for controlling access to the system memory, and a processor. *The network interface is configured for outputting data packets according to a prescribed flow control protocol that specifies flow control resources.* The network interface also is configured for ***outputting a data flow interruption request to the memory controller based on a determined depletion of the flow control resources***.
>
> (Page 2, line 29 to page 3, line 4).

---

[1]"During patent examination, the pending claims must be 'given their broadest reasonable interpretation consistent with the specification.'" MPEP §2111 at 2100-46 (Rev. 3, Aug. 2005) (*quoting In re Hyatt*, 211 F.3d 1367, 1372, 54 USPQ2d 1664, 1667 (Fed. Cir. 2000)).

"The broadest reasonable interpretation of the claims must also be consistent with the interpretation that those skilled in the art would reach." MPEP §2111.01 at 2100-47 (Rev. 3, Aug. 2005) (*citing In re Cortright*, 165 F.3d 1353, 1359, 49 USPQ2d 1464, 1468 (Fed. Cir. 1999)).

In addition, the specification explicitly states that packet transmission relies on the availability of flow control credits:

> [T]he [Host Channel Adapter] HCA 120 outputs the data packets onto assigned virtual lanes having respective flow control credits. However if the InfiniBand network encounters congestion in the form of a depletion of network bandwidth, indicated by ***a depletion of flow control credits (i.e., flow control resources)*** for a corresponding virtual lane, the HCA 120 must halt transmission of the data stream on the affected virtual lane, regardless of whether the data packets have already been generated.
>
> (Page 4, lines 27-30).
>
> The link layer control module 72 outputs the transmit packets according to a credit-based flow control. In particular, the link layer control module 72 monitors the available credits for transmission of a transmit packet on the assignment virtual lane. In particular, ***credits are sent on a per virtual lane basis, where a receiver issues a credit based on packets taken from an incoming virtual lane buffer; the credits are sent to the sender, enabling the sender to manage flow control***. Hence, ***if the link layer control module 72 determines that an identified virtual lane has an insufficient number of credits, the link layer control module 72 defers transmission of the corresponding transmit packet until a sufficient number of credits have been received, and sends the data flow interruption request to prevent generation of additional data packets***, as described above with respect to Figure 3. If the virtual lane has a sufficient number of credits, the link layer control module 72 forwards the transmit packet to the MAC module 46 for transmission, and if necessary outputs the resume data flow request as described above with respect to Figure 3.
>
> (Page 10, lines 20-31).

The specification also specifies that "the network interface (i.e., the HCA 120) detects a ***depletion of flow control credits*** for an identified virtual lane below a prescribed threshold. The network interface 120 in response outputs in step 210 ***a data flow interruption request to the memory controller 130*** for the identified virtual lane" (page 5, lines 16-18). Hence, the specification distinguishes between ***flow control credits*** and the ***data flow interruption request***.

The specification also explicitly states that "[t]he memory controller 130, in response to receiving the data flow interruption request, ***restricts in step 220 access by the processor 110 to system memory*** for the identified virtual lane (service level)" (page 5, lines 19-21); and that

"[t]he processor 110, in response to detecting in step 230 the unavailability of the system memory 48 for the identified virtual lane (or the identified service level), ***halts in step 240 the execution of the application resources*** utilizing the identified virtual lane"(page 5, lines 22-24).

Hence, the broadest reasonable interpretation of "depletion flow control resources" and "data flow interruption request" cannot be inconsistent with the specification, which requires that the "flow control resources" identify the availability of network bandwidth based on receipt of flow control information (e.g., flow control credits) from a receiver node via the (InfiniBand™) network, and that the "data flow interruption request" be distinct from the flow control information sent between the claimed "network node" and the receiver node via the network.

The prior art also provides a specific interpretation of the claimed "flow control resources": attached is Exhibit A of the Response After Final filed September 21, 2005 (pages 41, 44, and 175-176 of the InfiniBand™ Architecture Specification Release 1.0 in effect as of the filing date of the subject application and cited in the subject application). Also note that the InfiniBand™ Architecture Specification Release 1.0 is incorporated by reference in Pekkala et al. (para. 44).

As shown on the attached page 175 of the InfiniBand™ Architecture Specification (Exhibit A, page 3), "link level flow control [is used] to prevent the loss of packets due to buffer overflow by the receiver ***at each end of a link***." Further, page 175 of the InfiniBand™ Architecture Specification explicitly states (in para. 2 of 7.9.1) that "[t]he transmitter is the ***node*** sourcing data packets" and that "IBA receivers provide a 'credit limit' [that is] an indication of ***the total amount of data that the transmitter has been authorized to send*** since link utilization." (See para. 3 of 7.9.1).

Hence, it is notoriously well known in the art that flow control *packets* are used to control data flow between link partners at ***each end of the link*** based on the receiver sending a flow control "credit" to the transmitter..

In fact, Pekkala et al. also uses flow control credits to control data flow between link partners (i.e., network nodes) at ***each end of the link***: Fig. 1 of Pekkala et al. illustrates a network architecture having a switch 106, hosts 102 (each having an HCA 104) that are connected by

InfiniBand™ serial links 132 (see paragraph 44): Fig. 6 of Pekkala et al. describes an improved switch 106 that includes a transaction switch 602, and InfiniBand™ ports 608 (paragraphs 71, 75). These same InfiniBand™ ports 608 are illustrated in Fig. 7 as **connected to InfiniBand™ links 132** for connection with a **link partner 752** (paragraphs 82-86).

Pekkala et al. also describes in detail the use of flow control credits in paragraphs 60-67, where a flow control packet includes the number of flow control credits:

> [0060] Thus upon receiving a flow control packet 500 from its link partner, a port 208 may determine ***the amount of IB blocks worth of data packets 300 the port 208 is authorized to transmit in the specified VL 506***. That is, the port 208 may determine from the flow control packet 500 the amount of IB blocks worth of buffer space advertised by the link partner for the specified VL 506 ***according to the IBA specification incorporated by reference above***.
>
> [0061] Advertising zero IB blocks worth of credits, i.e., zero credits, instructs the link partner to stop transmitting data packets 300 in the specified VL 506. Advertising 66 IB blocks worth of credits, for example, authorizes the link partner to transmit one maximum-sized IB data packet 300 in the specified VL 506, i.e., 66 blocks*64 bytes/block=4224 bytes.

Pekkala et al. also describes with respect to Fig. 7 in paragraph 83 that "[t]he port 608 [of the switch 700] comprises an IB transmitter 724 that transmits IB packets, *such as data packets* ***300 and flow control packets 500, across one half of the full-duplex ID link 132 to a receiver 702 in the link partner 752***."

Hence, Pekkala et al. simply describes that the flow control logic 706 "stops" the transmitter 704 in response to receiving the flow control packets 500 ***from the port 608 of the switch via the InfiniBand link 132***:

> The link partner 752 [of Fig. 7] also includes flow control logic 706 coupled to the receiver 702 and transmitter 704. The link partner 752 flow control logic 706 receives flow control packets 500 from the link partner 752 ***receiver 702*** and provides flow control packets 500 to the link partner 752 transmitter 704. Among other things, the link partner 752 flow control logic 706 responds to flow control packets 500 **received from the port 608 advertising zero credits, and *responsively stops the link partner 752 transmitter 704 from transmitting IB data packets* 300 to the port 608**.

(Paragraph 84, lines 1-10 of Pekkala et al.).

Hence, Applicant's specification, Pekkala et al., *and* the Infiniband™ Architecture Specification (cited and relied on in both the subject application and Pekkala et al.) consistently describe flow control resources in terms of a receiver sending flow control packets to advertise available flow credits for sending a prescribed number of blocks of data.

Consequently, the broadest reasonable interpretation cannot be inconsistent with the specification and the interpretation those skilled in the art would reach, namely that the "depletion of flow control resources" refers to depletion of flow control credits as advertised by the receiver via the InfiniBand™ network, and that the "data flow interruption request" is distinct from the flow control messages specifying the flow control credits.

The Examiner presents an unreasonable interpretation of the claims by asserting on page 2 of the Final Action that "Pekkala does teach the feature of outputting by the network interface in the network node a data flow interruption request as shown in paragraphs 0084, 0092-0094, 0112 [*after detecting that there is no buffers* [sic] *are available to receive the data packet, issuing and providing a notification/flow* ***control packet*** *(i.e., outputting a* ***data flow interruption request****).*

As demonstrated above, the claims, the specification, Pekkala et al., *and* the InfiniBand™ Architecture specification consistently distinguish the flow control resources (i.e., flow control credits) from the claimed data flow interruption request. Moreover, the Examiner has failed to present any evidence whatsoever for any rational basis that one skilled in the art would have equated "data flow interruption request" with a "flow control packet", especially in view of the overwhelming evidence that consistently demonstrates that the claimed data flow interruption request is distinct from the flow control packets specifying the flow control credits.

In fact, the Final Action admits in paragraph 5 on page 3 that Pekkala sends ***packets*** to ***the link partner;*** hence, the Examiner's interpretation of "data flow interruption request" is therefore inconsistent with the teachings of Pekkala et al. when considered in its entirety, since the processor that is in the *same network node as the network interface generating the data flow*

interruption request reduces the prescribed data stream based on the data flow interruption request (See Argument A3 *infra*).

Hence, the §102 rejection of claims 1 and 7 should be reversed because it fails to demonstrate that the applied reference discloses each and every element of the claim, and because it disregards the explicit teachings of Pekkala et al.: while Pekkala et al. describes the credit-based flow control (which reads on the claimed flow control resources), Pekkala et al. does not disclose or suggest the claimed "data flow interruption request.[2]

A2. Pekkala et al. Does Not Disclose or Suggest the Claimed Reducing The Data Stream Based on Reducing Execution of a Prescribed Application Resource Based on the Data Flow Interruption Request

Independent claims 1 and 7 each specify that the processor (distinct from the network interface that generates the data flow interruption request) reduces "the prescribed data stream by ***reducing execution of a prescribed application resource***" that is configured for generating the prescribed data stream. Claims 1 and 7 also specify that the processor reduces execution of the prescribed application resource "based on the data flow interruption request."

The broadest reasonable interpretation of "reducing execution of a prescribed application resource" cannot be inconsistent with the specification, which specifies that actual generation of the packets is reduced or halted, based on reducing or halting execution of the actual application resource that generates the packets, eliminating the prior art problem of halting transmission of packets that have already been generated (see, e.g., page 2, lines 23-26 quoted on Page 7 *supra*; page 3 at lines 22-24: "network congestion can be resolved without the necessity of wasting resources by dropping packets generated by the processor and stored in a system memory"; page

---

[2] "Anticipation requires the presence in a single prior art reference disclosure of each and every element of the claimed invention, arranged as in the claim." *Lindemann Maschinenfabrik GmbH v. American Hoist & Derrick Co.*, 221 USPQ 481, 485 (Fed. Cir. 1984). Hence, it is not sufficient that a single prior art reference discloses each element that is claimed, but the reference also must disclose that the elements are arranged as in the claims under review. *In re Bond*, 15 USPQ2d 1566, 1567 (Fed. Cir. 1990) (citing *Lindemann Maschinenfabrik GmbH*).

11, lines 23-26).

The specification also explicitly states that "[t]he memory controller 130, in response to receiving the data flow interruption request, ***restricts in step 220 access by the processor 110 to system memory*** for the identified virtual lane (service level)" (page 5, lines 19-21); and that "[t]he processor 110, in response to detecting in step 230 the unavailability of the system memory 48 for the identified virtual lane (or the identified service level), ***halts in step 240 the execution of the application resources*** utilizing the identified virtual lane" (page 5, lines 22-24).

Hence, the broadest reasonable interpretation cannot be inconsistent with the specification, which requires reducing the prescribed data stream not only by restricting *transmission* of the data stream onto the network (in accordance with the depletion of the flow control resources), *but also* to reduce the actual generation of the prescribed data stream by the processor "***reducing execution*** of the prescribed application resource."[3]

Pekkala et al. provides no disclosure or suggestion whatsoever for how a transmitting network node should respond internally to a depletion of flow control credits. Rather, Pekkala et al. simply describes that the flow control logic in a receiver "stops" the link partner transmitter from **transmitting** packets to the destination port:

> Thus, no more free buffers will be available to receive incoming packets. In this case, the incoming ports must employ link level flow control to *stop their link partners from* ***transmitting packets*** until additional buffers become free.
>
> (Para. 12, lines 12-16)
>
> [0073] Consequently, the present inventors have advantageously observed that the amount of port buffering resources that are necessary need only be large enough to receive as many packets as the link partner can transmit, until the port can *stop the link partner from* ***transmitting any more packets***. This is independent of the particular virtual lanes specified in the transmitted packets.
>
> Advantageously, the size of an inline spill buffer 612 is sufficient to store packets 300 received during a latency period required to shut down the corresponding link partner ***from transmitting more packets*** 300....

---

[3]See Footnote 1 on Page 6 *supra*.

(Para. 76, lines 5-8)

The link partner 752 [of Fig. 7] also includes flow control logic 706 coupled to the receiver 702 and transmitter 704. The link partner 752 flow control logic 706 receives flow control packets 500 from the link partner 752 receiver 702 and provides flow control packets 500 to the link partner 752 transmitter 704. Among other things, **the link partner 752 flow control logic 706 responds to flow control packets 500 received from the port 608 advertising zero credits, and *responsively stops the link partner 752 transmitter 704 from transmitting IB data packets* 300 to the port 608**.

(Para. 84, lines 1-10)

The flow control packets 500 advertise zero credits in order to shut down the link partner 752 ***from transmitting data packets*** 300.

(Para. 96, lines 4-6)

Hence, Pekkala et al. simply describes that the flow control logic 706 responds to the depletion of flow control credits by halting transmission, and neither discloses nor suggests ***reducing execution of the prescribed application resource that generates the prescribed data stream***, as claimed.

The Examiner's tortured interpretation on page 3 of the Final Action of the halting of the link partner as teaching the claimed "reducing ***execution***" has no foundation: Pekkala et al. simply requires that the **transmitter 704 stop transmitting data packets**: there is no disclosure or suggestion whatsoever as to reducing ***execution of the prescribed application resource that generates those packets***. Hence, Pekkala et al. permits the link partner to continue execution of the application resource and continued generation of the data packets, so long as the packets are not transmitted.

Thus, the Examiner's assertion that the halting transmission of packets already having been generated is a teaching of the claimed reducing the prescribed data stream by ***reducing execution*** of the prescribed application resource configured for ***generating*** the prescribed data stream is unfounded and defies common sense. There is no teaching or suggestion in Pekkala et al. that the actual ***generation*** of the data stream should be restricted by reducing execution of the

prescribed application resource.[4]

Further, there is no inherency in reducing execution of the prescribed application resource, because all that is required in Pekkala et al. is *halting the transmission* of the packets that have ***already been generated,*** as opposed to the claimed reducing execution of the prescribed application resource that generates the prescribed data stream[5]; in fact, Pekkala et al. teaches that packets can be stored for transmission upon the receipt of a sufficient number of flow control credits:

> Thus, even if the link partner 752 has one flow control block (i.e., 64 bytes) of flow control credit, it cannot transmit a portion of a packet 300 waiting to be transmitted. Instead, the link partner 752 must wait until it has enough flow control credits to transmit an entire packet.
>
> (Para. 84, lines 12-17.

For these and other reasons, the §102 rejection should be reversed.

A3. Pekkala et al. Does Not Disclose or Suggest the Claimed Detecting Depletion, Outputting a Data Flow Interruption Request, and Reducing Execution of a Prescribed Application Resource Within The *Same Network Node*

A3.a. Claims 1 and 7 Explicitly Require The Claimed Operations Performed in The Same Network Node

Each of the independent claims 1 and 7 specify reducing a prescribed data stream in a network node by reducing execution of a prescribed application resource configured for

---

[4]See MPEP 2131. "The identical invention must be shown in as complete detail as is contained in the ... claim." *Richardson v. Suzuki Motor Co.*, 868 F.2d 1226, 1236, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989). "Anticipation cannot be predicated on teachings in the reference which are vague or based on conjecture." *Studiengesellschaft Kohle mbH v. Dart Industries, Inc.*, 549 F. Supp. 716, 216 USPQ 381 (D. Del. 1982), *aff'd.*, 726 F.2d 724, 220 USPQ 841 (Fed. Cir. 1984).

[5]When the Examiner alleges that a certain result is inherent in the operation of a reference, it must appear that this is necessarily so without any doubt. Ex parte Ruskin, 95USPQ 96 (Pat. Ofc. Bd. App. 1951).

generating the prescribed data stream. In particular, each independent claim specifies that each of the features of detecting a depletion of flow control resources, outputting a data flow interruption request based on the detected depletion of flow control resources, and reducing execution of the prescribed application resource that generates the prescribed data stream, ***are performed in the same network node***.

Specifically, claim 1 specifies a method "***in a network node***", the method including "detecting by a network interface ***in the network node*** a depletion of flow control resources", "outputting ***by the network interface*** a data flow interruption request based on the detected depletion of flow control resources", and "reducing, by a processor ***in the network node*** and *based on* ***the data flow interruption request***, the prescribed data stream by reducing execution of a prescribed application resource configured for generating the prescribed data stream."

Antecedent basis for "the network interface" in the step of "outputting by the network interface" is established by the recital of "a network interface ***in the network node***"; hence, the same network interface that is in the network node: detects the depletion of flow control resources, and outputs the data flow interruption request.

Further, the processor ***in the network node*** reduces the prescribed data stream by reducing execution of the prescribed application resource, "based on ***the*** data flow interruption request": antecedent basis is established by "outputting ***by the network interface*** [in the *network node*] a data flow interruption request".

Hence, the body of claim 1 explicitly requires that the claimed "detecting by a network interface in the network node", outputting "by a network interface a data flow interruption request", and "reducing, by a processor in the network node", be performed in the ***same network node***.

Independent claim 7 is an apparatus claim that specifies in its preamble "A network node" that comprises "a network interface" and "a processor". The claimed "network node" is a specific structural limitation that limits the structure of the claimed subject matter. Hence, the

recital of "a network node" <u>must</u> be treated as a claim limitation and cannot be disregarded.[6]

In fact, the specification <u>explicitly and consistently describes</u> that the processor (110 of Fig. 2), the network interface (120 of Fig. 2), the memory controller (130 of Fig. 2), and the system memory (48 of Fig. 2) reside in a <u>single network node</u> that communicates with other network nodes via the InfiniBand™ network.

Consequently, the recital in claim 7 of "a network node ***comprising***: a network interface ... and a processor" constitutes an <u>explicit requirement</u> that the network interface and the processor ***reside in the same network node***, namely the "network node" specified in the preamble.

A.3.b **The Claimed "Network Node" Cannot Be Disregarded**

The Examiner is deliberately disregarding the claimed feature that the detecting by the network interface, outputting by the network interface, and reducing by the processor is performed in "a network node", i.e., ***the same network node***:

> the features upon which applicant relies on (i.e., the data flow interruption request is output from the network interface to a destination ***within the same network node***, and that the processor ***within the same network node*** [sic]) are not recited in the rejected claim(s).

(Advisory Action, page 2, para. 2, emphasis in original).

As demonstrated above, the claims <u>explicitly require</u> that: the network interface and the processor are "in the network node" (claim 1), and that "a network node" ***comprises*** a network interface and a processor (claim 7); the network interface outputs the data flow interruption

---

[6]As acknowledged in §2111.02 of the MPEP, "[a]ny terminology in the preamble that limits the structure of the claimed invention must be treated as a claim limitation." MPEP, Rev. 3, Aug. 2005, p. 2100-51 (*citing Corning Glass Works v. Sumitomo Elec. U.S.A., Inc.*, 868 F.2d 1251, 1257, 9 USPQ2d 1962, 1966 (Fed. Cir. 1989)("The determination of whether preamble recitations are structural limitations can be resolved only on review of the entirety of the application 'to gain an understanding of what the inventors actually invented and intended to encompass in the claim.'"); *Pac-Tec Inc. v. Amerace Corp.*, 903 F.2d 796, 801, 14 USPQ2d 1871, 1876 (Fed. Cir. 1990); *In re Stencel*, 828 F.2d 751, 4 USPQ2d 1071 (Fed. Cir. 1987)).

request based on detecting the depletion of flow control resources; and the processor reduces the prescribed data stream based on the data flow interruption request by reducing execution of the prescribed application resource generating the data stream.

Hence, the rejection of independent claims 1 and 7 should be withdrawn because the deliberate disregard by the Examiner of the claimed "network node" is per se improper.[7]

Further, the Examiner cannot argue a "broadest reasonable interpretation" that somehow encloses a "network" *within* the network node: the broadest *reasonable* interpretation must be (1) consistent with the specification, and (2) consistent with the interpretation that those skilled in the art would reach.

As described in the specification, the claimed network node is illustrated in Fig. 2 (see page 4, lines 9-11 and 19-21) and includes a network interface (e.g., HCA 120), a memory controller 130, system memory 48, and a processor 110 that executes application resources 140a and 140b. The specification explicitly describes that a Host Channel Adapter (HCA) (e.g., 12 of Fig. 1 or 120 of Fig. 2) provides an interface connection to the InfiniBand™ network (e.g., 10 of Fig. 1) (see, e.g., page 2, lines 5-9, page 4, lines 27-31).

Hence, the term "network node" refers to an apparatus that is connected to a network link such as an InfiniBand™ network link. This interpretation also is consistent with the InfiniBand™ Architecture Specification, cited in the subject application: as shown on page 44 of the InfiniBand™ Architecture Specification (Page 2 of attached Exhibit A), a "Link" is defined as "[a] full duplex transmission path between ***any two network fabric elements*** [i.e., network nodes], such as Channel Adapters or Switches"; page 41 (Page 1 of attached Exhibit A) defines "Channel Adapter" as "[d]evice that ***terminates a link*** and executes transport-level functions. One of Host Channel Adapter or Target Channel Adapter." Further, page 175 of the InfiniBand™ Architecture Specification explicitly states that "[t]he transmitter is the ***node*** sourcing data packets."

---

[7]"All words in a claim must be considered in judging the patentability of that claim against the prior art." *In re Wilson*, 424 F.2d 1382, 1385, 165 USPQ 494, 496 (CCPA 1970)."

Therefore, any apparatus that <u>includes</u> an InfiniBand™ link is <u>not</u> a "network node", as claimed.

In fact, the applied reference Pekkala et al. incorporates by reference InfiniBand™ Architecture Specification, and further recognizes that each network node (i.e., an "end node") 102 includes a network interface (IB HCA 104), and that the end nodes 102 are connected by the InfiniBand™ links 132 (see, e.g., para. 44-45).

Consequently, any transfer of information via an InfiniBand™ link is <u>not</u> a transfer of information ***within the same network node***, as claimed.

A.3.c. Pekkala et al. Does Not Disclose Detecting Depletion, Outputting a Data Flow Interruption Request, and Reducing Execution of a Prescribed Application Resource Within The *Same Network Node*

The Examiner has the burden of establishing that Pekkala et al. discloses <u>each and every</u> element of the claim such that the identical invention must be shown in as complete detail as is contained in the claim.[8] Further, Further, anticipation cannot be established based on a piecemeal application of the reference, where the Examiner picks and chooses isolated features of the reference in an attempt to synthesize the claimed invention.[9] In other words, it is not sufficient that a single prior art reference discloses each element that is claimed, but the reference

---

[8]As specified in MPEP §2131: "'A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference' *Verdegaal Bros. V. Union Oil Co. of California,* 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987). ... 'The identical invention must be shown in as complete detail as is contained in the ... claim." *Richardson v. Suzuki Motor Co.*, 868 F.2d 1226, 1236, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989)." MPEP 2131 (Rev. 3, Aug. 2005, at p. 2100-76).

[9] "Anticipation requires the presence in a single prior art reference disclosure of each and every element of the claimed invention, <u>arranged as in the claim</u>." *Lindemann Maschinenfabrik GmbH v. American Hoist & Derrick Co.*, 221 USPQ 481, 485 (Fed. Cir. 1984). "Anticipation cannot be predicated on teachings in the reference which are vague or based on conjecture." *Studiengesellschaft Kohle mbH v. Dart Industries, Inc.*, 549 F. Supp. 716, 216 USPQ 381 (D. Del. 1982), *aff'd.*, 726 F.2d 724, 220 USPQ 841 (Fed. Cir. 1984).

also must disclose that the elements are arranged as in the claims under review. *In re Bond*, 15 USPQ2d 1566, 1567 (Fed. Cir. 1990) (citing *Lindemann Maschinenfabrik GmbH*).

In other words, the Examiner has the burden of establishing not only that Pekkala et al. discloses detecting a depletion of flow control resources by a network interface, outputting a data flow interruption signal by the network interface, and reducing by a processor the prescribed data stream (by reducing execution of the prescribed application resource) based on the data flow interruption request, but also that each and every one of the claimed operations be performed in the "network node", i.e., the same network node.

As demonstrated above with respect to Arguments A1 and A2, Pekkala et al. does not disclose or suggest performing these operations in the "network node", as claimed. Rather, Pekkala et al. discloses flow control between ***link partners***, i.e., distinct network nodes that transfer flow control credits via the InfiniBand links 132 in a System Area Network 100 (see, e.g., Fig. 1 and para. 44). There is no rational basis for equating the disclosed System Area Network 100 as a description of the claimed network node.

For these and other reasons, the §102 rejection of independent claims 1 and 7 should be withdrawn.

B. **Pekkala et al. Does Not Disclose or Suggest the Memory Controller of Dependent Claims 3, 8, and 13**

Each of the claims 3, 8, and 13 specify a memory controller configured for rendering unavailable the system memory resources for the prescribed application resource in response to reception of the data flow interruption request. As described in the specification:

> The memory controller 130, in response to reception of the data flow interruption request, renders unavailable the system memory resources 48 for the application resource (e.g., A1) associated with the affected virtual lane (VL1); application resources serviced by other virtual lanes (e.g., A2 serviced by VL2) remain unaffected, enabling the continued data flow of the application resources serviced by other virtual lanes. The processor 110, [in] response to detecting the unavailability of the system memory resources 48 for the corresponding application resource (A1), reduces the prescribed data stream by reducing execution of the corresponding application resource (A1), for example by temporarily

> suspending execution until reception of a resume data flow request indicating replenishment of the flow control resources for the prescribed data stream (VL1).
>
> (Page 5, lines 3-11)

Hence, claims 3, 8, and 13 specify that access to system memory resources are <u>restricted</u> in response to the data flow interruption request. Hence, the memory controller is able to cause the processor to reduce execution of the prescribed application resource to reduce the prescribed data stream, without adversely affecting other application resources. These and other features are neither disclosed nor suggested by Pekkala et al.

Paragraphs 85 and 94 of Pekkala et al do <u>not</u> disclose or suggest the claimed memory controller. Rather, these paragraphs simply describe a buffer control logic 606 within a transaction switch 602 that causes a received packet to be stored in an inline spill buffer 612, instead of the shared buffers 604, based on congestion in the shared buffers:

> When the buffer control logic 606 determines that a shared buffer 604 is not available to store a packet 300 received by the receiver 722, the buffer control logic 606 asserts the control signal 742 to **cause the inline spill buffer 612 to store the packet 300 data rather than passing the data through to the shared buffers 604**.
>
> (Para. 85, lines 8-13).
>
> The buffer control logic 606 attempts to allocate a buffer 604 for the packet 300 and determines no buffer 604 is available. The buffer control logic 606 notifies the flow control logic 726 via signal 744 to shutdown the link partner 752. The flow control logic 726 instructs the transmitter 724 to transmit zero credit flow control packets 500.
>
> (Para. 94, lines 7-14).

Hence, Pekkala et al. fails to disclose or suggest rendering unavailable *system* memory resources for the ***prescribed application resource***, but rather simply discloses the buffer control logic using a spill buffer 612, instead of the congested shared buffers 604, to store a received packet.

Pekkala et al. <u>also</u> teaches the <u>exact opposite</u> of rendering unavailable the system memory

resources "in response to ***reception of*** the data flow interruption request", by causing the flow control logic 726 to generate the flow control signal based on *determining that no buffer 604 is available*. In other words, the buffer control logic 606, in response to determining that no buffer 604 is available, sends a signal 744 that causes the flow control logic 726 to send the flow control packet.

In contrast, the claims specify that the memory controller renders unavailable the system memory resources ***in response to reception of the data flow interruption request*** generated by the network interface.

Hence, Pekkala et al. fails to disclose or suggest the memory controller of claims 3, 8, and 13; for these and other reasons, the §102 rejection of these claims should be reversed.

Conclusion

For the reasons set forth above, it is clear that Appellant's claims 1-13 are patentable over the reference applied. Accordingly the appealed claims 1-13 should be deemed patentable over the applied reference. It is respectfully requested that this appeal be granted and that the Examiner's rejections be reversed.

To the extent necessary, Appellant petitions for an extension of time under 37 C.F.R. 1.136 and 37 C.F.R. 41.37(e). Please charge any shortage in fees due in connection with the filing of this paper, including any missing or insufficient fees under 37 C.F.R. 1.17(a) or 41.20(b)(2), to Deposit Account No. 50-0687, under Order No. 95-508, and please credit any excess fees to such deposit account.

Respectfully submitted,

Manelli Denison & Selter, PLLC

Leon R. Turkevich
Registration No. 34,035

Customer No. 20736

CLAIMS APPENDIX

1. (PREVIOUSLY PRESENTED) A method in a network node, the method comprising:

detecting by a network interface in the network node a depletion of flow control resources representing a depletion of network bandwidth for a prescribed data stream;

outputting by the network interface a data flow interruption request based on the detected depletion of flow control resources; and

reducing, by a processor in the network node and based on the data flow interruption request, the prescribed data stream by reducing execution of a prescribed application resource configured for generating the prescribed data stream.

2. (PREVIOUSLY PRESENTED) The method of claim 1, wherein the network interface is configured for outputting the prescribed data stream according to InfiniBand™ protocol, the detecting step including detecting depletion of flow control credits, as the flow control resources, for a prescribed virtual lane.

3. (ORIGINAL) The method of claim 2, wherein the outputting step includes outputting the data flow interruption request to a memory controller configured for controlling access to system memory resources, the memory controller rendering unavailable the system memory resources for the prescribed application resource in response to reception of the data flow interruption request.

4. (ORIGINAL) The method of claim 3, wherein the reducing step includes halting execution of the prescribed application resource, based on a determined unavailability of the system memory resources.

5. (ORIGINAL) The method of claim 4, further comprising outputting by the network interface a resume data flow request based on a detected replenishment of the flow control

resources for the prescribed data stream.

6. (ORIGINAL) The method of claim 5, further comprising resuming execution of the prescribed application resource based on the resume data flow request.

7. (ORIGINAL) A network node comprising:

a network interface configured for detecting a depletion of flow control resources representing a depletion of network bandwidth for a prescribed data stream, the network interface configured for outputting a data flow interruption request based on the detected depletion of flow control resources; and

a processor configured for executing a prescribed application resource for generation of the prescribed data stream, the processor configured for reducing the prescribed data stream by reducing execution of the prescribed application resource, based on the data flow interruption request.

8. (ORIGINAL) The network node of claim 7, further comprising a memory controller configured for controlling access to system memory resources, the memory controller configured for rendering unavailable the system memory resources for the prescribed application resource in response to reception of the data flow interruption request.

9. (ORIGINAL) The network node of claim 8, wherein the processor is configured for reducing the execution of the prescribed application resource based on detecting the unavailability of the system memory resources.

10. (ORIGINAL) The network node of claim 9, wherein the network interface is configured for outputting a resume data flow request based on a detected replenishment of the flow control resources for the prescribed data stream.

11. (ORIGINAL) The network node of claim 10, wherein the processor is configured for resuming execution of the prescribed application resource based on the resume data flow request.

12. (PREVIOUSLY PRESENTED) The network node of claim 7, wherein the network interface is configured for outputting the prescribed data stream according to InfiniBand™ protocol, the network interface configured for detecting depletion of flow control credits, as the flow control resources, for a prescribed virtual lane.

13. (PREVIOUSLY PRESENTED) The method of claim 1, wherein the outputting step includes outputting the data flow interruption request to a memory controller in the network node and that is configured for controlling access to system memory resources in the network node, the memory controller rendering unavailable the system memory resources for the prescribed application resource in response to reception of the data flow interruption request.

EVIDENCE APPENDIX –

Attached: 4 Pages of Exhibit A of Response After Final filed September 21, 2005

CA — See Channel Adapter.

Channel — The association of two queue pairs for communication.

Channel Adapter — Device that terminates a link and executes transport-level functions. One of Host Channel Adapter or Target Channel Adapter.

Channel Interface — The presentation of the channel to the Verbs Consumer as implemented through the combination of the Host Channel Adapter, associated firmware, and device driver software.

Channel, Reliable Datagram — See Reliable Datagram Channel.

CI — See Channel Interface.

Client — The active entity in an active/passive communication establishment exchange.

CM — See Communication Manager.

CME — Chassis Management Entity.

Communication Manager — The software, hardware, or combination of the two that supports the communication management mechanisms and protocols.

Completion Error — Permanent interface or processing error reported through completion status.

Completion Queue — A queue containing one or more Completion Queue Entries. Completion Queues are internal to the Channel Interface, and are not visible to verb consumers.

Completion Queue Entry — The Channel Interface-internal representation of a Work Completion.

Connection — An association between a pair of entities (e.g., processes) over one or more Channels.

Consumer — See Verbs Consumer.

CQE — Completion Queue Entry, commonly pronounced "cookie".

CRC — Cyclic Redundancy Check.

Data Payload — The data, not including any control or header information, carried in one packet.

Data Segment — A tuple in a Work Request that specifies a virtually contiguous buffer for Host Channel Adapter access. Each Data Segment consists of a Virtual

Exhibit A, Response After Final filed September 21, 2005, Appln. No. 09/905,080
Page 1 of 4

IPv6	Internet Protocol, version 6
IPv6 Address	A 128-bit address constructed in accordance with IETF RFC 2460 for IPv6.
Key	A construct used to limit access to one or more resources, similar to a password. The following keys are defined by the InfiniBand™ Architecture: Baseboard Management Key Local Key Management Key Queue Key Partition Key Remote Key
L_Key	See Local Key.
LID	See Local Identifier.
LID Mask Control	A per-port value assigned by the Subnet Manager. The value of the LMC specifies the number of Path Bits in the Local Identifier.
Link	A full duplex transmission path between any two network fabric elements, such as Channel Adapters or Switches.
LMC	See LID Mask Control.
Local Identifier	An address assigned to a port by the Subnet Manager, unique within the subnet, used for directing packets within the subnet. The Source and Destination LIDs are present in the Local Route Header. A Local Identifier is formed by the sum of the Base LID and the value of the Path Bits.
Local Key	An opaque object, created by a verb, referring to a Memory Region, used with a Virtual Address to describe authorization for the HCA hardware to access local memory. It may also be used by the HCA hardware to identify the appropriate page tables for use in translating virtual to physical addresses.
Local Route Header	Routing header present in all InfiniBand™ Architecture packets, used for routing through switches within a subnet.
Local Subnet	The collection of links and Switches that connect the Channel Adapters of a particular subnet.
LRH	See Local Route Header.

Table 32 Link Packet

Field	Value
FCCL	0x21B

Generated FCCRC: 0xF9C9

Table 33 Link Packet Byte Stream

Byte	Value
0	0x01
1	0x0D
2	0x52
3	0x1B
4	0xF9
5	0xC9

7.9 Flow Control

7.9.1 Introduction

This section describes the link level flow control mechanism utilized by IBA to prevent the loss of packets due to buffer overflow by the receiver at each end of a link. This mechanism does not describe end to end flow control such as might be utilized to prevent transmission of messages during periods when receive buffers are not posted.

Throughout this section, the terms "transmitter" and "receiver" are utilized to describe each end of a given link. The transmitter is the node sourcing data packets. The receiver is the consumer of the data packets. Each end of the link has a transmitter and a receiver.

IBA utilizes an "absolute" credit based flow control scheme. Unlike many traditional flow control schemes which provide incremental updates that are added to the transmitters available buffer pool, IBA receivers provide a "credit limit". A credit limit is an indication of the total amount of data that the transmitter has been authorized to send since link initialization.

Errors in transmission, in data packets, or in the exchange of flow control information can result in inconsistencies in the flow control state perceived by the transmitter and receiver. The IBA flow control mechanism provides

for recovery from this condition. The transmitter periodically sends an indication of the total amount of data that it has sent since link initialization. The receiver uses this data to re-synchronize the state between the receiver and transmitter.

7.9.2 FLOW CONTROL BLOCKS

The term “flow control block”, or simply “block” indicates a quantity of data in a data packet. This quantity is defined to be the size of the data packet in bytes (every byte between the local route header and the variant CRC, inclusive) divided by 64 bytes, and rounded up to the next integral value.

7.9.3 RELATIONSHIP TO VIRTUAL LANES

The flow control algorithm defined in this chapter is applied to each virtual lane independently, except for virtual lane 15 which is not subject to link level flow control.

7.9.4 FLOW CONTROL PACKET

Figure 60 Flow Control Packet Format

<table>
<tr><th colspan="9">Flow Control Packet - general format</th></tr>
<tr><th>bits
bytes</th><th colspan="2">31-24</th><th colspan="2">23-16</th><th colspan="2">15-8</th><th colspan="2">7-0</th></tr>
<tr><td>0-3</td><td>Op</td><td colspan="3">FCTBS</td><td>VL</td><td colspan="3">FCCL</td></tr>
<tr><td>4-5</td><td colspan="4">LPCRC</td><td colspan="4"></td></tr>
</table>

C7-53: Flow control packets shall be sent for each VL except VL15 upon entering the LinkInitialize state. When in the PortStates LinkInitialize, LinkArm or LinkActive, a flow control packet for a given virtual lane shall be transmitted prior to the passing of 65,536 symbol times since the last time a flow control packet for the given virtual lane was transmitted.

C7-54: Flow control packets shall use the format specified in Figure 60 Flow Control Packet Format on page 176.

A symbol time is defined as the time required to transmit an eight bit data quantity onto the link. Flow control packets may be transmitted as often as necessary to return credits and enable efficient utilization of the link. See Section 7.6.4, “Buffering and Flow Control For Data VLs,” on page 149 for additional information.

7.9.4.1 FLOW CONTROL PACKET FIELDS

7.9.4.1.1 OPERAND (OP) - 4 BITS

The flow control packet is a link packet with one of two Op (operand) values: An operand of 0x0 indicates a normal flow control packet. An operand value of 0x1 indicates a flow control init packet.